Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

ADA207065

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FJSRL-TR-89-0003	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Frank J. Seiler Research Lab 6b. OFFICE SYMBOL (If applicable) FJSRL/NC	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) USAF Academy Colorado 80840-6528	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Office of Scientific Research 8b. OFFICE SYMBOL (If applicable) AFOSR	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Bldg 410 Bolling AFB DC 20332	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2303; TASK NO. F3; WORK UNIT ACCESSION NO. 05

11. TITLE (Include Security Classification) ESR Base: An Electron Spin Resonance Spectrum Baseline Correction Routine Using Chebyschev Polynominals (Unclassified)

12. PERSONAL AUTHOR(S)
Joseph A. Menapace

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical Report	FROM Nov 88 TO Mar 89	1989/March	68

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
07	04		Electron Spin Resonance, Least-Squares Fit, Chebyschev Polynomials, Baseline Correction,
20	05		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

An electron spin resonance (ESR) spectrum baseline correction routine is presented for use in modeling/correcting experimental ESR spectral data. The routine corrects for baseline intensity offset and baseline intensity drift by approximating the baseline using a Chebyschev (power-series) polynomial expansion. The correction routine's source code is written in FORTRAN 77 and accepts computerized ESR spectral data as input from files written in ASCII format. After correcting the baseline to zero intensity, the routine outputs ASCII files containing the corrected ESR spectral data, the baseline intensity data, and the Chebyschev polynomial coefficients for use in subsequent analyses. The algorithm used to implement the Chebyschev polynomial baseline correction and the utilities needed to perform the calculation are described and serve as a guide to the user for both routine operation and subsequent modification of the computer source code to a particular application.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Joseph A. Menapace, Capt, USAF	22b. TELEPHONE (Include Area Code) (719)472-2655 22c. OFFICE SYMBOL FJSRL/NC

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

ESR BASE

An Electron Spin Resonance
Spectrum Baseline Correction Routine
Using Chebyschev Polynomials

Version 1.00

1989

Joseph A. Menapace, Capt, USAF

Frank J. Seiler Research Laboratory
FJSRL/NC
United States Air Force Academy
Colorado 80840–6528
Phone: (719) 472–2655

COPYRIGHT STATUS

ABSTRACT

An electron spin resonance (ESR) spectrum baseline correction routine is presented for use in modeling/correcting experimental ESR spectral data. The routine corrects for baseline intensity offset and baseline intensity drift by approximating the baseline using a Chebyschev (power–series) polynomial expansion. The correction routine's source code is written in FORTRAN 77 and accepts computerized ESR spectral data as input from files written in ASCII format. After correcting the baseline to zero intensity, the routine outputs ASCII files containing the corrected ESR spectral data, the baseline intensity data, and the Chebyschev polynomial coefficients for use in subsequent analyses. The algorithm used to implement the Chebyschev polynomial baseline correction and the utilities needed to perform the calculation are described and serve as a guide to the user for both routine operation and subsequent modification of the computer source code to a particular application.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont)

INTRODUCTION

Computer control of various electron spin resonance (ESR) spectrometer functions and automated acquisition and manipulation of ESR spectra make a significant impact on the speed, efficiency, and effectiveness of data collection and analysis. ESR spectra collected via computer, however, are anything but ideal. Particularly, baseline offset or baseline drift can occur which could either be induced by the operator or by certain experimental and instrumental parameters. These artifacts make it extremely difficult and cumbersome to analyze the spectra by other computerized techniques such as numerical integration, numerical differentiation, numerical maximum/minimum locating, subtraction and addition of spectra, etc., as well as making it difficult to compare experimental spectra with simulated spectra. The solution to this problem sets the cornerstone for the computer program discussed in this report. The subject program, ESRBASE, performs a baseline correction to computerized ESR spectral data utilizing Chebyschev polynomials and numerical Least–Squares techniques.

DESCRIPTION OF THE ALGORITHM

The methodology used in the implementation of Least–Squares techniques to the baseline correction routine is presented with two goals in mind. First, the section will give the reader the necessary background to understand the Least–Squares techniques involved in performing the baseline correction. Second, it will serve as a basis for documenting the technique so that subsequent modifications/tailoring of the computer code by others can be made with a high degree of certainty and ease.

METHOD OF LEAST–SQUARES:

If one considers the ESR spectral baseline to be, in general, a line possessing some amount of slowly varying curvature, an approximation can be made to model the baseline intensity utilizing Chebyschev (power–series) polynomials[1]:

1) $$y = \sum_{j=0}^{n} a_j x^j$$

for which the dependent variable, in this case the baseline intensity y, is written as a function of the independent variable, the magnetic field strength or field position x, and constant coefficients, a_j. The summation runs over the number of coefficients which corresponds to the order of the polynomial n plus one.

The application of Least–Squares techniques[2] to the Chebyschev polynomial baseline model to determine the constant coefficients, a_j, involves the minimization of the sum of squares of deviations between the experimental y data and the y values calculated using Equation 1:

2) $$\sum_{i=1}^{m} (\Delta y_i)^2 = \sum_{i=1}^{m} [(y_i - \sum_{j=0}^{n} a_j x_i^j)^2]$$

for which Δy_i is the difference between the ith observed y value (m observations) and the ith

"calculated" y value. To determine the constant coefficients, a_j, and hence the optimum fit to the baseline, the sum of squares of the deviations must be minimized. This is done by taking partial derivatives of Equation 2 with respect to each of the constant coefficients:

3)
$$\frac{\delta(\Delta y_i)^2}{\delta a_0} = -2\sum_{i=1}^{m}[(y_i - \sum_{j=0}^{n} a_j x_i^j)]$$

$$\frac{\delta(\Delta y_i)^2}{\delta a_1} = -2\sum_{i=1}^{m}[(y_i - \sum_{j=0}^{n} a_j x_i^j)x_i]$$

$$\frac{\delta(\Delta y_i)^2}{\delta a_k} = -2\sum_{i=1}^{m}[(y_i - \sum_{j=0}^{n} a_j x_i^j)x_i^k]$$

$$\vdots \qquad\qquad \vdots$$

$$\frac{\delta(\Delta y_i)^2}{\delta a_n} = -2\sum_{i=1}^{m}[(y_i - \sum_{j=0}^{n} a_j x_i^j)x_i^n]$$

for which k=0, n. Setting the derivatives equal to zero and rearranging yields a set of n+1 simultaneous equations:

4)
$$\sum_{i=1}^{m} y_i = \sum_{i=1}^{m}\sum_{j=0}^{n} a_j x_i^j$$

$$\sum_{i=1}^{m} x_i y_i = \sum_{i=1}^{m} x_i \sum_{j=0}^{n} a_j x_i^j = \sum_{i=1}^{m}\sum_{j=0}^{n} a_j x_i^{(j+1)}$$

$$\sum_{i=1}^{m} x_i^k y_i = \sum_{i=1}^{m} x_i^k \sum_{j=0}^{n} a_j x_i^j = \sum_{i=1}^{m}\sum_{j=0}^{n} a_j x_i^{(j+k)}$$

$$\vdots \qquad\qquad \vdots$$

$$\sum_{i=1}^{m} x_i^n y_i = \sum_{i=1}^{m} x_i^n \sum_{j=0}^{n} a_j x_i^j = \sum_{i=1}^{m} \sum_{j=0}^{n} a_j x_i^{(j+n)}$$

for which k=0, *n*. Equations 4 are solved to give the constant coefficients, a_j. The coefficient values obtained are the constant coefficients for which the sum of squares of the deviations between the experimental data points and the calculated Chebyschev polynomial fit form a minimum.

The solution to the set of simultaneous equations (Equations 4) can be obtained in a variety of different ways. The method chosen involves matrix inversion techniques. Here, the solution for the coefficients of the n+1 simultaneous equations is found by left multiplying the matrix form of Equations 4 (Equation 5) by the matrix inverse of $\underline{X}$ (Equation 6):

5) $$\underline{X}\underline{a}=\underline{b}$$

for which:

$$b_k = \sum_{i=1}^{m} x_i^k y_i$$

a_k=kth order polynomial coefficient

$$X_{jk} = \sum_{i=1}^{m} x_i^{(j+k)}$$

6) $$\underline{X}^{-1}\underline{b}=\underline{X}^{-1}\underline{X}\underline{a}=\underline{a}$$

The matrix inverse $\underline{X}^{-1}$ utilized in Equation 6 is obtained using the Gauss–Jordan matrix inversion technique[3-4]. The technique involves performing identical elimination operations on the matrix, $\underline{X}$, and on the unit matrix, $\underline{1}$. Since $\underline{X}^{-1}$ can be written as:

7) $$\underline{X}^{-1} = \underline{1}/\underline{X}$$

the identical elimination operations will not change the ratio and will result in the unit matrix becoming $\underline{X}^{-1}$ and the matrix $\underline{X}$ becoming the unit matrix. The elimination operations used in the inversion process are taken from the operations used in the Doolittle Scheme[5]:

8) $$X'_{jk} = X_{jk} - X_{ik}\left(\frac{X_{ji}}{X_{ii}}\right)$$

for which i is the index for the row being retained. For example, if Equation 8 is applied to all rows except the first in a matrix, all the elements of the first column will vanish except X_{11}. This equation essentially normalizes the row of the matrix to insure that it will cause some of the matrix elements from another row to vanish when the normalized row is subtracted from the row of interest. The normalization in the situation at hand is given as X_{ji}/X_{ii} which corresponds to the portion of the elimination operation which is common to both the matrix, $\underline{X}$, and the unit matrix so that the unit matrix becomes $\underline{X}^{-1}$ and that $\underline{X}$ becomes the unit matrix. The Gauss–Jordan matrix inversion procedure used in the ESRBASE algorithm involves application of Equation 8 to the matrix $\underline{X}$ and to the unit matrix followed by division of the retained row by the corresponding diagonal element of $\underline{X}$. This procedure is repeated until the matrix $\underline{X}$ is reduced to the unit matrix. Once this occurs, the unit matrix is transformed into $\underline{X}^{-1}$.

PROGRAM DESCRIPTION

The computer code comprising ESRBASE is composed of eleven routines written in FORTRAN 77 with some special features peculiar to Microsoft Corporation's implementation of the language in their version of MS–FORTRAN 4.1. The reason for utilizing MS–FORTRAN 4.1 coding is that ESRBASE is intended for use on IBM personal computers and compatibles. However, the source code can be modified for use on other computer systems as well with the modifications outlined in the Installation Section of this report. The eleven FORTRAN routines have the following file names:

ESRBASE.FOR
SPECIN.FOR
SELECT.FOR
SCALE.FOR
CHEFIT.FOR
INVERS.FOR
UNSCAL.FOR
CHEOUT.FOR
SPECHE.FOR
SPECOT.FOR
DIMSIZES.DAT

The purpose/function of each routine is described below. The source code for the routines is reproduced in Appendix A so the reader may reference the appropriate sections as he reads through the descriptions.

ESRBASE.FOR:

ESRBASE.FOR comprises the driver for the baseline correction routine. It controls the sequence of program execution and sets up the necessary user input parameters for utilization in the program subroutines. The routine displays the maximum data point pair and polynomial degree allocations so the user knows what size ESR spectral data files can be used in the fit as well as the size of the largest polynomial degree. The degree of the

Chebyschev polynomial to be used in the fit is input from this routine by the user.

ESRBASE uses four files during computation which are described as follows. The input data file FOR005 appearing in the routine is a generic file name used for assignment to DATIN. The file FOR005 must contain the user supplied magnetic field strength (field position) and ESR signal intensity data pairs to be used in the baseline correction in ASCII format. The output files FOR006, FOR007, and FOR008, appearing in the routine are generic file names used for assignment to DATOUT, DATLIS, and DATCOM, respectively. FOR006 contains the baseline corrected ESR spectrum composed of field position/ESR intensity data pairs (ASCII). FOR007 contains the listing of the Chebyschev polynomial coefficients determined from the correction. FOR008 contains the original ESR spectrum composed of field position/ESR intensity data pairs along with the best fit baseline intensity as a third column entry (ASCII). FOR008 provides a sort of check on how well the baseline correction functions if the field position/ESR intensity data pairs are plotted and then overlaid with a plot of the field position/best fit baseline intensity data pairs. If desired, the user could modify these file name assignments to suit his particular application; for example, the user may wish to incorporate ESRBASE as a subroutine in a larger ESR spectral analysis package.

SPECIN.FOR:

The subroutine SPECIN.FOR performs the input of file FOR005 into the baseline correction routine. The file must contain magnetic field strength/ESR intensity data entered as pairs. For example, one line of the file would look like:

-30.00,1000.0

or

-30.00 1000.0

for which the first entry is the magnetic field strength and the second is the ESR signal intensity. The subroutine also determines the number of data point pairs contained in the file and terminates the correction routine if the maximum allocation of data point pairs is exceeded. An input data file with a different file name than FOR005 must be copied to FOR005 if the executable ESRBASE.EXE is used to perform the baseline correction; otherwise, the file must be named with a .DAT extension if the BASELINE.BAT batch file (described in Program Operation Section) is used to perform the baseline correction.

SELECT.FOR:

The subroutine SELECT.FOR selects the portion of the ESR baseline intensity to be used in the baseline correction. SELECT.FOR performs the selection by determining the mean baseline intensity and the standard deviation about the mean baseline intensity. The subroutine outputs the number of standard deviations needed to encompass all the ESR signal intensity data as well as the number of standard deviations needed to encompass the wing baseline intensity to the screen. These values serve as guides to the user in determining the number of standard deviations which are to be used in the baseline intensity selection. The user must input how many standard deviations which are to be used to select the baseline intensity for the fit. Once the number of standard deviations are entered, SELECT.FOR selects the portion of the ESR baseline intensity that falls within the limits set. The mean baseline intensity is substituted for those intensity values that are not within the limits.

SCALE.FOR:

The subroutine SCALE.FOR scales the ESR baseline data point pairs for the Chebyschev polynomial fitting. The subroutine accomplishes the scaling by determining the largest magnitudes of both the magnetic field strength and the ESR baseline signal intensity and then divides all the field position and ESR baseline intensity data points by these two values, respectively. The net result is data pairs with values ranging between $-1.0 \leq x \leq 1.0$ and $-1.0 \leq y \leq 1.0$. Scaling is performed in the baseline correction since large x and/or y values cause the matrix elements in Equation 5 to become extremely large, sometimes resulting in math overflow problems during matrix inversion.

CHEFIT.FOR:

The subroutine CHEFIT.FOR performs the polynomial fit to the ESR baseline data and determines the values of the constant coefficients in the Chebyschev polynomial. CHEFIT.FOR accomplishes the summations presented in Equation 5 then sets up the matrix to be used in the inversion calculation. Once the matrix inversion is accomplished, Equation 6 is used to calculate the values of the coefficients.

INVERS.FOR:

The subroutine INVERS.FOR determines the inverse of the square matrix $\underline{X}$ for subroutine CHEFIT.FOR. The subroutine calculates the matrix inverse via the Gauss–Jordan inversion technique outlined in the Algorithm Section (see Equations 7 and 8).

UNSCAL.FOR:

Subroutine UNSCAL.FOR scales the Chebyschev polynomial constant coefficients needed to perform the final baseline correction to the same domain encompassing the original magnetic field strength (field position) and ESR signal intensity.

CHEOUT.FOR:

The subroutine CHEOUT.FOR outputs the results of the Chebyschev polynomial baseline correction to output data file FOR007. The output consists of a listing of the Chebyschev polynomial coefficients calculated for an nth degree baseline correction. An example of a line output in FOR007 is:

$$a_1 = 1.2345678D+01$$

SPECHE.FOR:

Subroutine SPECHE.FOR outputs both the original ESR spectrum consisting of magnetic field strength/ESR signal intensity data pairs along with a third column consisting of the best fit baseline intensity corresponding to the Chebyschev polynomial baseline correction. These data are written to the output file FOR008 in ASCII format. FOR008 can serve as a check on how well the baseline correction functions if the field position/ESR intensity data pairs are plotted and then overlaid with a plot of the field position/best fit baseline intensity data pairs. An example of a line output in FOR008 is:

30.0000 , 1000.0 , 1000.0

SPECOT.FOR:

Subroutine SPECOT.FOR outputs the baseline corrected ESR spectrum as magnetic field strength/ESR signal intensity data pairs. These data are written to output file FOR006 in ASCII format. An example of a line output in FOR006 is:

30.0 , 1000.0

DIMSIZES.DAT:

DIMSIZES.DAT provides the parameters used in the ESRBASE routines to establish the maximum array dimensions necessary to accomplish a baseline correction. The relevant parameters include the maximum number of data pairs and the maximum degree of the Chebyschev polynomial that ESRBASE will be able to use to perform the baseline correction. The DIMSIZES.DAT file is introduced into the FORTRAN routines via the $INCLUDE: MS–FORTRAN 4.1 metacommand. The maximum number of data pairs default is 2000 data point pairs. The maximum Chebyschev polynomial degree default is 30. If the user desires to alter these parameters from their default settings, the procedure outlined in the Installation Section must be used.

INSTALLING ESRBASE

The computer code comprising ESRBASE is written in FORTRAN 77 and can be compiled using Microsoft Corporation's MS–FORTRAN version 4.1 or later. The reason for utilizing MS–FORTRAN 4.1 coding is that ESRBASE is intended for use on IBM–AT personal computers and compatibles. Furthermore, the installation routine coding is written for machines operating under either MS–DOS or PC–DOS versions 3.2 or later. If the user does not have MS–FORTRAN 4.1 and MS–DOS/PC–DOS 3.2 or later, the programs may need to be modified to operate correctly. It is advised that the user refer to the specific documentation for the FORTRAN compiler and/or operating system used on the system that will maintain/operate ESRBASE to accomplish this task.

Installation of the ESRBASE baseline correction routine onto a computer system can be conducted using the installation program INSTALL.BAT. The installation program copies all the FORTRAN source code routines described in the Program Description Section, a batch run utility, BASELINE.BAT, a compilation utility, COMPILE.BAT, and a test example, TESTA.DAT, into a subdirectory located on the computer system. BASELINE.BAT allows the user to run ESRBASE on a specific input file other than FOR005. This program will be discussed in the Running ESRBASE Section. COMPILE.BAT allows the user to recompile the baseline correction routine if the DIMSIZES.DAT file (or any other FORTRAN source code file) has been altered after the baseline correction routine has been initially installed. To use INSTALL.BAT, the target subdirectory MUST exist prior to the installation; otherwise, the installation will not occur. INSTALL.BAT is initiated by typing INSTALL followed by the target drive name and target subdirectory name. For example, the following command installs the baseline correction routine onto drive C in subdirectory UTILITY.ESR:

INSTALL C:\UTILITY.ESR

INSTALL.BAT subsequently compiles the FORTRAN routines using MS–FORTRAN 4.1 compilation calls. The compilation defaults are to compile using in–line floating point instructions for the 80287 math package (/FPi87) and the 80286 instruction set (/G2). Upon completing this step, the installation program links the object code generated in the compilation to create the executable ESRBASE.EXE. To complete the installation, INSTALL.BAT deletes the object codes generated in the compilation from the subdirectory.

Users may wish to alter the default parameters in DIMSIZES.DAT to create a baseline correction routine which accepts a different maximum number of data point pairs and/or a different maximum Chebyschev polynomial degree. This is accomplished by changing the parameter expressions, MAXPRS and/or MAXDGE in DIMSIZES.DAT. MAXPRS is the parameter name assigned which allocates the maximum number of data point pairs that ESRBASE will accept as input from file FOR005. MAXPRS must be assigned as the maximum number of data point pairs accepted plus one. MAXDGE is the parameter name assigned which allocates the maximum Chebyschev polynomial degree that ESRBASE will accept as a user input. MAXDGE must be assigned as the maximum polynomial degree plus one. For example, if the user desires to create a baseline correction routine which will accept a maximum of 1000 data point pairs and a maximum 10th degree polynomial, the parameters in DIMSIZES.DAT must have the following assignments:

MAXPRS=1001
MAXDGE=11

ESRBASE can be used on computer systems other than IBM PC–AT or compatibles or with other FORTRAN compilers than MS–FORTRAN 4.1 if modifications are made to the FORTRAN source code. Since the routines comprising the ESRBASE source code are written, for the most part, in FORTRAN 77, the only modifications necessary involve changing the coding peculiar to MS–FORTRAN 4.1. To modify ESRBASE to compile and run on other computer systems and using other than MS–FORTRAN 4.1, the user must change the metacommand $INCLUDE: to the appropriate include statement format accompanying the system and compiler used by the computer system/compiler where ESRBASE is to be maintained/operated. The installation program, compilation utility, and batch run utility (described in the Running ESRBASE section) will also have to be modified. The user is advised to refer to the specific documentation for the FORTRAN compiler and/or operating system used on the system that will maintain/operate ESRBASE to accomplish this task.

RUNNING ESRBASE

The ESRBASE baseline correction routine can be utilized two ways in performing baseline corrections to ESR spectral data. First, the executable ESRBASE.EXE can be used to perform the correction on a data file named FOR005; or second, a batch file named BASELINE.BAT can be used to perform the baseline correction on any data file possessing a .DAT file name extension.

Baseline corrections to ESR spectral data can be made by running ESRBASE.EXE directly. Since ESRBASE accepts raw ESR spectral data from file FOR005, the user must copy the file containing the spectral data to FOR005. At this point execution of ESRBASE.EXE will begin the baseline correction. Before the correction is accomplished, ESRBASE will prompt the user for the degree of the Chebyschev polynomial desired to correct the baseline as well as the number of standard deviations necessary to encompass the baseline intensity data.

Once the user has entered the necessary information, ESRBASE will perform the baseline correction and create three output files named FOR006, FOR007, and FOR008. FOR006 contains the baseline corrected ESR spectrum composed of field position/ESR intensity data pairs (ASCII). FOR007 contains the listing of the Chebyschev polynomial coefficients determined from the correction. FOR008 contains the original ESR spectrum composed of field position/ESR intensity data pairs along with the best fit baseline intensity as a third column entry (ASCII). These files must be renamed after the correction is complete in order to use ESRBASE.EXE to analyze other ESR spectra as the information will be overwritten in subsequent calculations.

BASELINE.BAT is a batch file designed to perform a baseline correction via ESRBASE.EXE which accepts any input data file name with a .DAT extension. The user, therefore, does not need to copy the specific ESR spectral data file to FOR005 in order to perform the baseline correction. BASELINE.BAT copies the specified data file to the input file as well as renames the output files to those which have the specific data file name as a base name plus unique extensions. For example, the file mapping which BASELINE.BAT accomplishes on an ESR data file named NITOX.DAT is:

NITOX.DAT → FOR005
FOR006 → NITOX.OUT
FOR007 → NITOX.LIS
FOR008 → NITOX.CHK

As shown, the output file FOR006 is renamed to the base data file name with the .OUT extension. FOR007 is renamed to the base data file name with the .LIS extension. FOR008 is renamed to the base data file name with the .CHK extension. Before performing the correction, BASELINE checks to see if the .DAT file exists. If the .DAT file does not exist, the program will be terminated with the message:

File was not found – File does not exist or was not specified

BASELINE.BAT also checks to see if output files exist with the same base data file name. If these files exist, BASELINE.BAT terminates with the message (using NITOX base file name as an example):

NITOX.OUT NITOX.LIS NITOX.CHK Already exist – Rename the NITOX.DAT file.

The BASELINE.BAT batch file is executed by typing BASELINE followed by the base data file name. For example, the following command performs the baseline correction on an ESR data file named NITOX.DAT:

BASELINE NITOX

A TEST EXAMPLE

A test example file named TESTA.DAT is included with the ESRBASE baseline correction routine to show the results of a typical baseline correction to ESR spectral data. The ESR spectrum contained in TESTA.DAT is shown in Figure 1. The spectrum is that of a nitroxyl radical formed via ultraviolet (250 nm) photolysis of a solution of perdeuterio–trinitrobenzene in perdeuterio–toluene at room temperature. The nitroxyl

radical spectrum was obtained at Frank J. Seiler Research Laboratory, USAF Academy, Colorado using a Perkin–Elmer E–109 Electron Spin Resonance Spectrometer. Even though the spectrum baseline is more or less flat and the signal–to–noise ratio is excellent, the baseline is offset about –80 intensity units from zero and possesses a slight baseline drift from low to high magnetic field strength.

The ESR spectrum is baseline corrected with ESRBASE by using two standard deviations about the mean baseline intensity and a tenth order Chebyschev polynomial. Figure 2 presents the results contained in the FOR008 file. The dotted line in the figure corresponds to the original ESR spectrum and the solid line corresponds to the Least–Squares fit to the baseline using the tenth order Chebyschev polynomial.

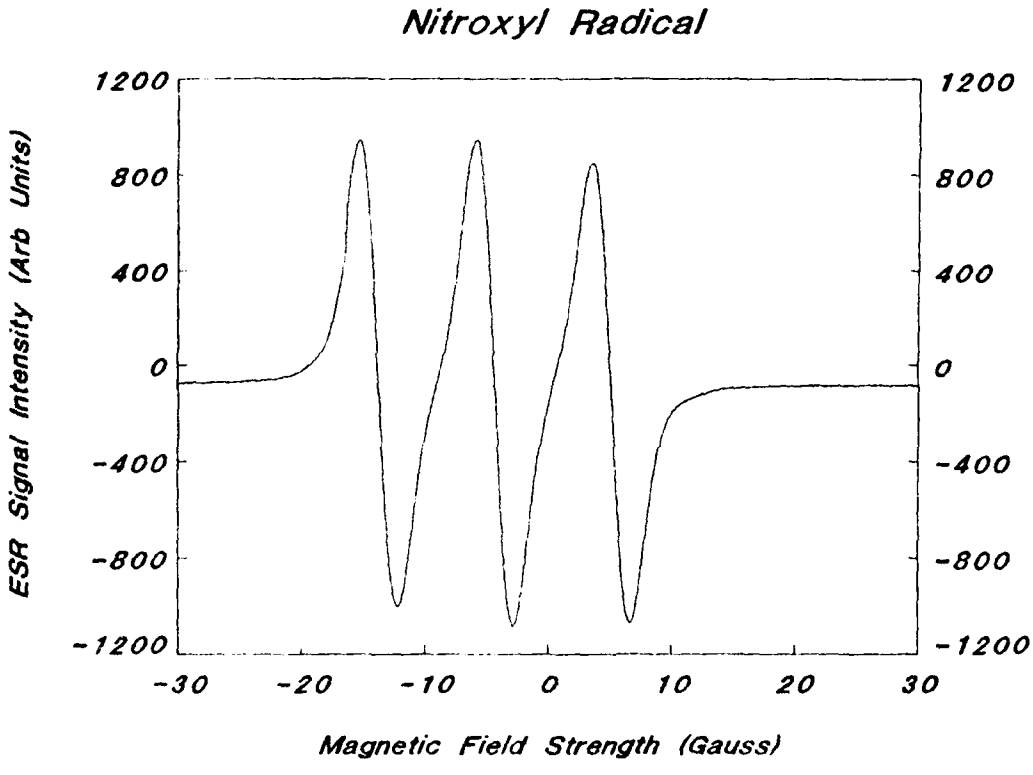

Figure 1: ESR Spectrum of nitroxyl radical to be baseline corrected using ESRBASE (File FOR005).

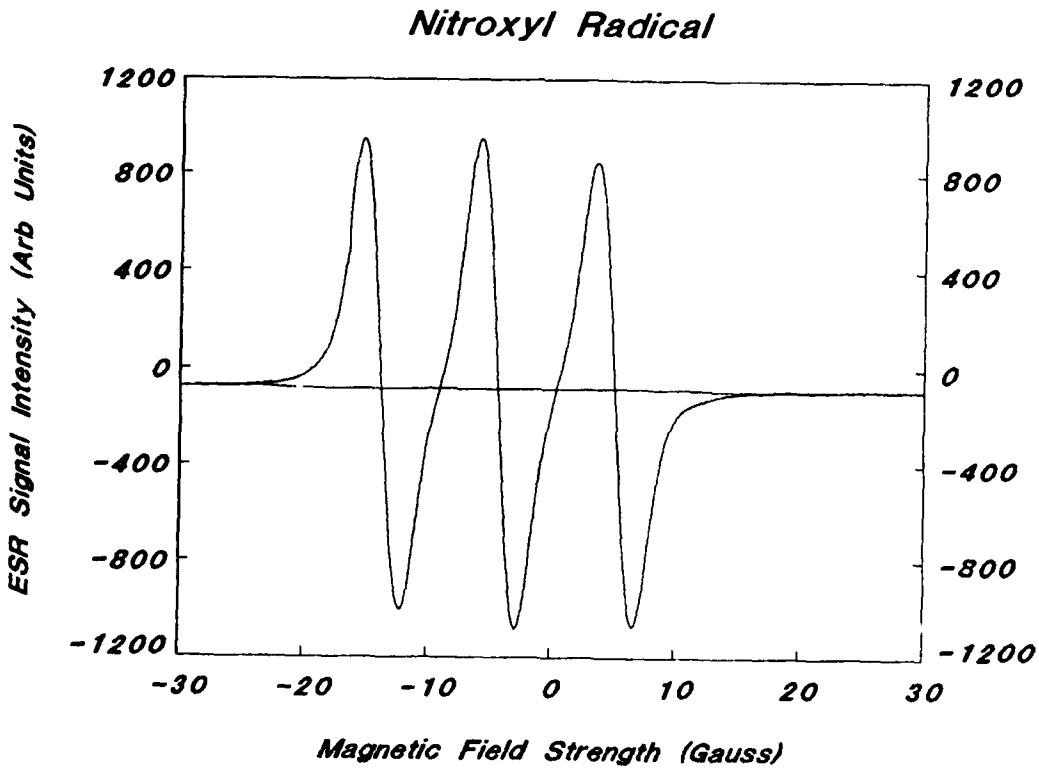

Figure 2: Least–Squares baseline fit to the nitroxyl radical spectrum corrected using ESRBASE (File: FOR008).

The Chebyschev polynomial coefficients obtained from the correction are (file FOR007):

Coefficients of the Chebyschev Polynomial:
where $y(x)=a_0+a_1x+a_2x^3+a_3x^3+...$

$a_0 =$ –8.222289700D+01
$a_1 =$ 1.774521785D–01
$a_2 =$ 6.437718754D–02
$a_3 =$ –3.386472903D–03
$a_4 =$ –8.160498804D–04
$a_5 =$ 1.047132670D–05
$a_6 =$ 2.757092729D–06
$a_7 =$ –1.476481618D–08
$a_8 =$ –3.505265709D–09
$a_9 =$ 7.674939038D–12
$a_{10} =$ 1.517105041D–12

The resulting corrected nitroxyl radical ESR spectrum is shown in Figure 3 (file FOR006).

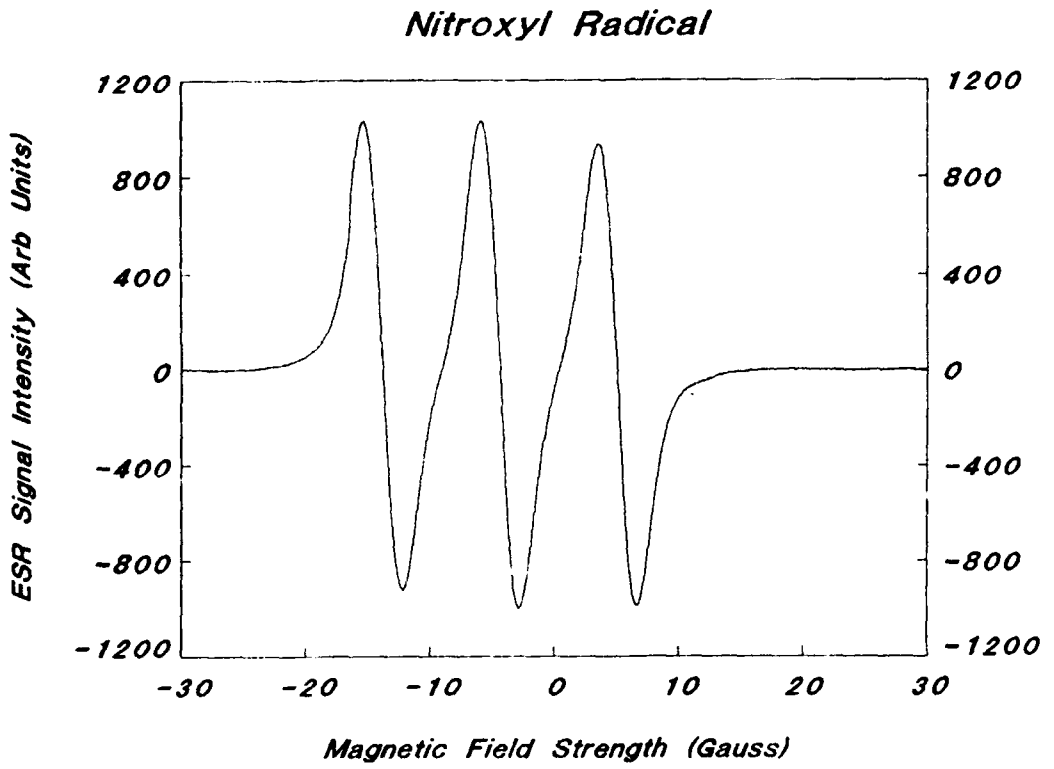

Figure 3: Baseline corrected nitroxyl radical ESR spectrum (File: FOR006).

REFERENCES

1. Arfken, G.," Mathematical Methods for Physicists", 3rd Edition, Orlando, Florida: Academic Press, Inc. (1985).

2. Draper, N.R. and H. Smith, "Applied Regression Analysis", 2nd Edition, New York: John Wiley & Sons, Inc. (1981).

3. Pennington, R.H., "Introductory Computer Methods and Numerical Analysis", New York: Macmillan Publishing Co, Inc. (1970).

4. Ralston, A., and H. Wilf, Eds, "Mathematical Methods for Digital Computers", New York: John Wiley & Sons, Inc. (1960).

5. Beyer, William H. Ed, "CRC Standard Mathematical Tables", 25th Edition, West Palm Beach, Florida: CRC Press, Inc. (1978).

Appendix A

ESRBASE FORTRAN and Utility Source Code

ESRBASE.FOR

```
C   PROGRAM ESRBASE
C
C   PURPOSE
C
C   PERFORM A LEAST-SQUARES FIT TO DATA USING A POLYNOMIAL
CURVE
C     Y=A(1)+A(2)*X+A(3)*X**2+A(4)*X**3+...
C     THE CORRECTION IS USED TO CORRECT THE BASELINE OF AN ESR
C     SPECTRUM USED AS THE INPUT DATA SET
C
C   DESCRIPTION OF PARAMETERS
C
C   XFIELD - ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C   YINTEN - ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C   NPAIRS - NUMBER OF DATA POINT PAIRS
C   NDEGRE - NUMBER OF COEFFICIENTS (DEGREE OF POLYNOMIAL+1)
C   POLYN  - ARRAY OF COEFFICIENTS OF POLYNOMIAL
C   I      - ITERATION COUNTER
C   MAXPRS - MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C   MAXDGE - MAXIMUM DEGREE OF POLYNOMIAL + 1
C   DATIN  - ESR SPECTRUM INPUT FILE FOR005
C   DATOUT - CORRECTED ESR SPECTRUM OUTPUT FILE FOR006
C   DATCOM - ESR SPECTRUM / BASELINE OUTPUT FILE FOR008
C   DATLIS - BASELINE CORRECTION RESULTS OUTPUT FILE FOR007
C   REDX   - SCALE FACTOR FOR INDEPENDENT VARIABLE
C   REDY   - SCALE FACTOR FOR DEPENDENT VARIABLE
C   XDATA  - RAW DATA:  ESR FIELD STRENGTH
C   YDATA  - RAW DATA:  ESR SIGNAL INTENSITY
C
C   SUBROUTINES REQUIRED
C
C   SPECIN(DATIN,NPAIRS)
C     INPUTS THE ESR SPECTRUM FROM FILE FOR005
C   SELECT(NPAIRS)
C     SELECTS THE PORTION OF THE ESR SPECTRUM TO BE USED IN THE
C     LEAST-SQUARES FIT
C   SCALE(NPAIRS,REDX,REDY)
C     NORMALIZES THE ESR SPECTRUM TO BE BASELINE CORRECTED
C   CHEFIT(NPAIRS,NDEGRE)
C     PERFORMS THE CHEBYCHEV POLYNOMIAL FIT TO THE DATA
C   UNSCAL(REDX,REDY,NDEGRE)
C     RESTORES NORMALIZED ESR SPECTRUM TO ITS ORIGINAL SIZE
C     AND RESCALES POLYNOMIAL COEFFICIENTS
C   SPECHE(NPAIRS,NDEGRE,DATCOM)
C     OUTPUTS THE ESR SPECTRUM AND BASELINE CORRECTION TO FILE
C     FOR008
C   SPECOT(NPAIRS,NDEGRE,DATOUT)
C     OUTPUTS THE BASELINE CORRECTED ESR SPECTRUM TO FILE FOR006
C   CHEOUT(DATLIS,NDEGRE)
C     OUTPUTS THE POLYNOMIAL FIT RESULTS TO THE SCREEN AND TO FILE
C     FOR007
C   INVERS(NDEGRE)
C     CALCULATES THE INVERSE OF A SYMMETRIC TWO-DIMENSIONAL
C     MATRIX OF ORDER NDEGRE
```

```
C
C    NOTE
C
C    THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C     ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      PROGRAM ESRBASE

      DOUBLE PRECISION XFIELD,YINTEN,POLYN,REDX,REDY,XDATA,YDATA
      INTEGER NPAIRS,NDEGRE,MAXPRS,MAXDGE,I
      CHARACTER*20 DATIN,DATOUT,DATLIS,DATCOM

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XFIELD(1:MAXPRS),YINTEN(1:MAXPRS),POLYN(1:MAXDGE),
     1 XDATA(1:MAXPRS),YDATA(1:MAXPRS)
      COMMON/BLK2/XFIELD,YINTEN
      COMMON/BLK3/XDATA,YDATA
      COMMON/BLK4/POLYN

      DATIN='FOR005'
      DATOUT='FOR006'
      DATLIS='FOR007'
      DATCOM='FOR008'
      WRITE (*,'(24(/))')
      WRITE (*,1)
1                                                                    FORMAT
('-------------------------------------------------------------------
     1-------------------------')
      WRITE (*,*) '                    E S R   B A S E'
      WRITE (*,*)
      WRITE (*,*) '                  Electron Spin Resonance'
      WRITE (*,*) '                Chebychev Polynomial Baseline
     1'
      WRITE (*,*) '                    Correction Routine'
      WRITE (*,*)
      WRITE (*,*) '                    Version 1.00  1989'
      WRITE (*,*)
      WRITE (*,*) '                         by'
      WRITE (*,*)
      WRITE (*,*) '                  Joseph A. Menapace, Capt, USA
     1F'
      WRITE (*,*) '               Frank J. Seiler Research Laborat
     1ory'
      WRITE (*,*) '                      FJSRL/NC'
      WRITE (*,*) '                United States Air Force Academ
     1y'
      WRITE (*,*) '              Colorado 80840    Phone: (719) 472
     1-2655'
      WRITE (*,1)
      WRITE (*,*)
      WRITE (*,5) 'Maximum data point pair allocation is ',MAXPRS-1,'.'
      WRITE (*,5) 'Maximum polynomial degree allocation is',MAXDGE-1,'.'
```

```
5    FORMAT (A41,I4,A1)
     WRITE (*,'(2(/))')
     WRITE (*,*) 'Press return to continue.'
     READ (*,*)
     WRITE (*,'(24(/))')
     CALL SPECIN(DATIN,NPAIRS)
     CALL SELECT(NPAIRS)
40   WRITE (*,1)
     WRITE (*,*) '                    Parameter Entry'
     WRITE (*,1)
     WRITE (*,*)
     WRITE (*,41) 'There are ',NPAIRS,' pairs of data points used in th
    1e correction.'
41   FORMAT (A11,I4,A45)
     WRITE (*,*)
     WRITE (*,*) 'Enter the degree of the polynomial approximation to b
    1e used in the correction.'
     READ (*,*) NDEGRE
     NDEGRE=NDEGRE+1
     WRITE (*,*)
     IF (NDEGRE.GE.1.AND.NDEGRE.LT.NPAIRS.AND.NDEGRE.LE.MAXDGE)
    1GO TO 50
     IF (NDEGRE.LT.1) WRITE (*,*) 'ERROR !! Polynomial degree must be
    1positive.'
     IF (NDEGRE.GE.NPAIRS) WRITE (*,42)'ERROR !! Polynomial degree mus
    1t be less than ',NPAIRS-1,'.'
     IF (NDEGRE.GT.MAXDGE) WRITE (*,42)'ERROR !! Polynomial degree mus
    1t be less than ',MAXDGE,'.'
42   FORMAT (A47,I4,A1)
     GO TO 40
50   WRITE (*,'(24(/))')
     CALL SCALE(NPAIRS,REDX,REDY)
     CALL CHEFIT(NPAIRS,NDEGRE)
     CALL UNSCAL(REDX,REDY,NDEGRE)
     CALL SPECHE(NPAIRS,NDEGRE,DATCOM)
     CALL SPECOT(NPAIRS,NDEGRE,DATOUT)
     CALL CHEOUT(DATLIS,NDEGRE)
     END
```

SPECIN.FOR

```
C    SUBROUTINE SPECIN
C
C    PURPOSE
C
C    INPUTS THE ESR SPECTRUM TO BE USED IN BASELINE CORRECTION
C
C    USAGE
C
C    CALL SPECIN(DATIN,NPAIRS)
C
C    DESCRIPTION OF PARAMETERS
C
C    DATIN  – ASSIGNED INPUT FILE FOR005
C    XDATA – ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C    YDATA – ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C    NPAIRS – NUMBER OF DATA POINT PAIRS
C    I      – ITERATION COUNTER
C    MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C    MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C
C    SUBROUTINES REQUIRED
C
C    NONE
C
C    NOTE
C
C    THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C     ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE SPECIN(DATIN,NPAIRS)

      INTEGER NPAIRS,I,MAXDGE,MAXPRS
      DOUBLE PRECISION XDATA,YDATA
      CHARACTER*20 DATIN

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XDATA(1:MAXPRS),YDATA(1:MAXPRS)
      COMMON/BLK3/XDATA,YDATA

      WRITE (*,1)
1                                                                    FORMAT
('-------------------------------------------------------------------
     1-------------------------')
      WRITE (*,*) '                    ESR Spectrum Input'
      WRITE (*,1)
      WRITE (*,*)
      OPEN (UNIT=5,FILE=DATIN,STATUS='OLD')
      NPAIRS=0

C
C    INPUTS XDATA AND YDATA ONLY
C
```

```
10   READ (5,*,END=20) XDATA(NPAIRS+1),YDATA(NPAIRS+1)
     NPAIRS=NPAIRS+1
     IF (NPAIRS.GT.MAXPRS) GO TO 100
     GO TO 10
20   CONTINUE
     CLOSE (UNIT=5,STATUS='KEEP')
     WRITE (*,'(26(/))')
     IF (NPAIRS.EQ.0) GO TO 200
     GO TO 1000
100  CLOSE (UNIT=5,STATUS='KEEP')
     WRITE (*,*) 'The data file is too large.'
     WRITE (*,*) 'Array dimensions must be increased in order to use ro
    1utine.'
     WRITE(*,*)
     STOP
200  WRITE (*,*) 'The ESR spectral data file is empty.'
     WRITE (*,*) 'Correction cannot be done.'
     WRITE (*,*) 'Calculation aborted.'
     STOP
1000 RETURN
     END
```

SELECT.FOR

```
C    SUBROUTINE SELECT
C
C    PURPOSE
C
C    SELECTS THE PORTION OF THE ESR SPECTRUM TO BE USED
C    IN THE BASELINE CORRECTION
C
C    USAGE
C
C    CALL SELECT(NPAIRS)
C
C    DESCRIPTION OF PARAMETERS
C
C    XFIELD – ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C    YINTEN – ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C    NPAIRS – NUMBER OF DATA POINT PAIRS
C    I      – ITERATION COUNTER
C    MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C    MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C    SUMY   – SUMMATION OF ESR INTENSITY DATA
C    SIGMA  – STANDARD DEVIATION OF THE ESR INTENSITY DATA
C    SIGMAM – STANDARD DEVIATION OF THE MEAN ESR INTENSITY
C    YMEAN  – MEAN ESR SIGNAL INTENSITY
C    FREE   – DEGREES OF FREEDOM
C    SUM    – NUMBER OF ESR FIELD STRENGTH DATA POINTS
C    CONFID – INTERVAL USED TO SELECT BASELINE PORTIONS
C    STDDEV – NUMBER OF STANDARD DEVIATION USED TO SELECT
BASELINE
C    UPPER  – UPPER INTERVAL END POINT
C    LOWER  – LOWER INTERVAL END POINT
C    XDATA  – RAW DATA: ESR FIELD STRENGTH DATA
C    YDATA  – RAW DATA: ESR SIGNAL INTENSITY DATA
C    MAXY   – LARGEST ABSOLUTE VALUE OF ESR SIGNAL INTENSITY
C    LSIGMA – LOW END STANDARD DEVIATION
C    HSIGMA – HIGH END STANDARD DEVIATION
C
C    SUBROUTINES REQUIRED
C
C    NONE
C
C    NOTE
C
C    THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C     ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

     SUBROUTINE SELECT(NPAIRS)

     DOUBLE                                                    PRECISION
XFIELD,YINTEN,SUMY,SIGMA,SIGMAM,YMEAN,FREE,SUM,
    1 CONFID,STDDEV,UPPER,LOWER,XDATA,YDATA,MAXY,LSIGMA,HSIGMA
     INTEGER NPAIRS,MAXPRS,MAXDGE,I

$INCLUDE:'DIMSIZES.DAT'
```

```
      DIMENSION XFIELD(1:MAXPRS),YINTEN(1:MAXPRS),XDATA(1:MAXPRS),
     1 YDATA(1:MAXPRS)
      COMMON/BLK2/XFIELD,YINTEN
      COMMON/BLK3/XDATA,YDATA

      WRITE (*,1)
1                                                                    FORMAT
('-------------------------------------------------------------------
     1-------------------------')
      WRITE (*,*) '                    ESR Baseline Selection'
      WRITE (*,1)
      WRITE (*,*)
      SUMY=0.D0
      SUM=0.D0
      SIGMA=0.D0
      MAXY=0.D0
      DO 10 I=1,NPAIRS
        IF (ABS(YDATA(I)).GT.MAXY) MAXY=ABS(YDATA(I))
        SUM=SUM+1.D0
        SUMY=SUMY+YDATA(I)
10    CONTINUE
      YMEAN=SUMY/SUM
      DO 20 I=1,NPAIRS
        SIGMA=SIGMA+(YDATA(I)-YMEAN)**2
20    CONTINUE
      FREE=NPAIRS-1
      IF (FREE.NE.0.D0) GO TO 15
      WRITE (*,*) 'Cannot select a ESR baseline due to lack of data.'
      WRITE (*,*) 'Calculation terminated.'
      STOP
15    SIGMA=SQRT(SIGMA/FREE)
      SIGMAM=SIGMA/SQRT(SUM)
      IF (SIGMAM.NE.0.D0) GO TO 16
      WRITE (*,*) 'The ESR spectrum is perfectly flat.'
      WRITE (*,*) 'Cannot determine a standard deviation for the data.'
      WRITE (*,*) 'Calculation terminated.'
      STOP
16    LSIGMA=ABS(YDATA(1)-YMEAN)/SIGMAM
      HSIGMA=ABS(YDATA(NPAIRS)-YMEAN)/SIGMAM
      IF (LSIGMA.GT.HSIGMA) HSIGMA=LSIGMA
      WRITE (*,21) 'The average baseline intensity is ',YMEAN,'.'
21    FORMAT (A35,F12.6,A1)
      WRITE (*,22)'The standard deviation of the average baseline intens
     1ity is ',SIGMAM,'.'
22    FORMAT (A61,F12.6,A1)
      WRITE (*,*)
      WRITE (*,*) 'To encompass all the data, the number of standard dev
     1iations'
      WRITE (*,23) 'should be at least ',MAXY/SIGMAM,'.'
23    FORMAT (A20,F12.6,A1)
      WRITE (*,*)
      WRITE (*,*) 'To encompass the wing baseline intensity, the number
     1of standard'
      WRITE (*,24) 'deviations should be at least ',HSIGMA,'.'
```

```
24   FORMAT (A31,F12.6,A1)
     WRITE (*,*)
     WRITE (*,*) 'Enter the number of standard deviations to be used in
    1 the correction.'
     READ (*,*) STDDEV
     CONFID=SIGMAM*STDDEV
     UPPER=YMEAN+CONFID
     LOWER=YMEAN-CONFID
     DO 30 I=1,NPAIRS
       IF (UPPER.LT.YDATA(I).OR.LOWER.GT.YDATA(I)) GO TO 25
         XFIELD(I)=XDATA(I)
         YINTEN(I)=YDATA(I)
         GO TO 30
25       XFIELD(I)=XDATA(I)
         YINTEN(I)=YMEAN
30   CONTINUE
     WRITE (*,'(26(/))')
     RETURN
     END
```

SCALE.FOR

```
C    SUBROUTINE SCALE
C
C    PURPOSE
C
C     SCALES (NORMALIZES) ESR SPECTRUM FOR POLYNOMIAL FIT
C
C    USAGE
C
C    CALL SCALE (NPAIRS,REDX,REDY)
C
C    DESCRIPTION OF PARAMETERS
C
C     XFIELD – ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C     YINTEN – ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C     NPAIRS – NUMBER OF DATA POINT PAIRS
C     REDX   – SCALING (NORMALIZING) FOR INDEPENDENT VARIABLE
C     REDY   – SCALING (NORMALIZING) FOR DEPENDENT VARIABLE
C     I      – ITERATION COUNTER
C     MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS
C     MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C
C    SUBROUTINES AND FUNCTION PROGRAMS REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C       ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE SCALE (NPAIRS,REDX,REDY)

      DOUBLE PRECISION XFIELD,YINTEN,REDX,REDY
      INTEGER NPAIRS,I,MAXPRS,MAXDGE

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XFIELD(1:MAXPRS),YINTEN(1:MAXPRS)
      COMMON/BLK2/XFIELD,YINTEN

      REDX=1.D0
      REDY=1.D0
      WRITE (*,*) 'Scaling the esr spectrum for the calculations.'
      WRITE (*,*)

C
C     DETERMINES LARGEST XFIELD AND YINTEN
C

      DO 10 I=1,NPAIRS
        IF (ABS(XFIELD(I)).GT.REDX) REDX=ABS(XFIELD(I))
        IF (ABS(YINTEN(I)).GT.REDY) REDY=ABS(YINTEN(I))
10    CONTINUE
```

```
C
C    NORMALIZES XFIELD AND YINTEN
C

     DO 20 I=1,NPAIRS
       XFIELD(I)=XFIELD(I)/REDX
       YINTEN(I)=YINTEN(I)/REDY
20   CONTINUE
     RETURN
     END
```

CHEFIT.FOR

```
C    SUBROUTINE CHEFIT
C
C    PURPOSE
C
C     PERFORM A LEAST-SQUARES FIT TO DATA WITH A POLYNOMIAL CURVE
C       Y=A(1)+A(2)*X+A(3)*X**2+A(4)*X**3+...
C
C    USAGE
C
C     CALL CHEFIT(NPAIRS,NDEGRE)
C
C    DESCRIPTION OF PARAMETERS
C
C     XFIELD - ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C     YINTEN - ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C     NPAIRS - NUMBER OF DATA POINT PAIRS
C     NDEGRE - NUMBER OF COEFFICIENTS (DEGREE OF POLYNOMIAL+1)
C     POLYN  - ARRAY OF COEFFICIENTS OF POLYNOMIAL
C     I,J,K  - ITERATION COUNTERS
C     MAXPRS - MAXIMUM NUMBER OF DATAPOINT PAIRS + 1
C     MAXDGE - MAXIMUM DEGREE OF POLYNOMIAL + 1
C     ARRAY  - MATRIX FOR INVERSION TO DETERMINE POLYNOMIAL
C           COEFFICIENTS
C     XI     - INTERMEDIATE INDEPENDENT VARIABLE
C     YI     - INTERMEDIATE DEPENDENT VARIABLE
C     SUMX   - WEIGHTED SUM OF INDEPENDENT VARIABLE
C     SUMY   - WEIGHTED SUM OF DEPENDENT VARIABLE
C     XTERM  - INTERMEDIATE WEIGHTING FOR SUM OVER X'S
C     YTERM  - INTERMEDIATE WEIGHTING FOR SUM OVER Y'S
C     NMAX   - INDEPENDENT VARIABLE NUMBER MAXIMUM FOR WEIGHTED
SUM
C           OVER INDEPENDENT VARIABLE
C
C    SUBROUTINES REQUIRED
C
C     INVERS(NDEGRE)
C       CALCULATES THE INVERSE OF A SYMMETRIC TWO-DIMENSIONAL
C       MATRIX OF ORDER NDEGRE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C       ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE CHEFIT(NPAIRS,NDEGRE)

      DOUBLE PRECISION XFIELD,YINTEN,POLYN,SUMX,SUMY,
     1 ARRAY,XI,YI,XTERM,YTERM,DELTA
      INTEGER NPAIRS,NDEGRE,NMAX,J,K,I,MAXPRS,MAXDGE

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XFIELD(1:MAXPRS),YINTEN(1:MAXPRS),POLYN(1:MAXDGE),
```

```
     1 SUMX(1:2*MAXDGE-1),SUMY(1:MAXDGE),
     2 ARRAY(1:MAXDGE,1:MAXDGE)
      COMMON/BLK1/ARRAY
      COMMON/BLK2/XFIELD,YINTEN
      COMMON/BLK4/POLYN

C
C     ACCUMULATE WEIGHTED SUMS
C

      WRITE (*,*) 'Accumulating sums of the data.'
      WRITE (*,*)
      NMAX=2*NDEGRE-1
      DO 13 I=1,NMAX
        SUMX(I)=0.D0
13    CONTINUE
      DO 15 I=1,NDEGRE
        POLYN(I)=0.D0
        SUMY(I)=0.D0
15    CONTINUE
      DO 50 I=1,NPAIRS
        XI=XFIELD(I)
        YI=YINTEN(I)
        XTERM=1.D0
        DO 44 J=1,NMAX
          SUMX(J)=SUMX(J)+XTERM
          XTERM=XTERM*XI
44      CONTINUE
        YTERM=YI
        DO 48 J=1,NDEGRE
          SUMY(J)=SUMY(J)+YTERM
          YTERM=YTERM*XI
48      CONTINUE
50    CONTINUE

C
C     CONSTRUCT MATRICES AND CALCULATE COEFFICIENTS
C

      WRITE (*,*) 'Constructing the coefficient matrix.'
      WRITE (*,*)
      DO 54 I=1,NDEGRE
        DO 55 J=1,NDEGRE
          K=I+J-1
          ARRAY(I,J)=SUMX(K)
55      CONTINUE
54    CONTINUE
      CALL INVERS(NDEGRE)
      WRITE (*,*) 'Calculating the Chebyschev polynomial coefficients.'
      WRITE (*,*)
      DO 60 I=1,NDEGRE
        DO 70 J=1,NDEGRE
          POLYN(I)=POLYN(I)+ARRAY(I,J)*SUMY(J)
70      CONTINUE
```

```
60   CONTINUE
80   RETURN
     END
```

INVERS.FOR

```
C    SUBROUTINE INVERS
C
C    PURPOSE
C
C     CALCULATES THE INVERSE OF A SYMMETRIC TWO-DIMENSIONAL
C     MATRIX OF ORDER NDEGRE
C
C    USAGE
C
C     CALL INVERS(NPAIRS)
C
C    DESCRIPTION OF PARAMETERS
C
C     I,J,K  - ITERATION COUNTER
C      L
C     MAXDGE - MAXIMUM DEGREE OF POLYNOMIAL + 1
C     MAXPRS - MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C     ARRAY  - ARRAY FOR INVERSION TO DETERMINE THE POLYNOMIAL
C           COEFFICIENTS
C     AMAX   - LARGEST ARRAY ELEMENT IN MATRIX ARRAY
C     SAVE   - TEMPORARY ARRAY ELEMENT STORAGE
C     IK     - ROW INTERCHANGE INDEX
C     JK     - COLUMN INTERCHANGE INDEX
C     NDEGRE - NUMBER OF POLYNOMIAL COEFFICIENTS (DEGREE + 1)
C
C    SUBROUTINES REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C      ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE INVERS(NDEGRE)

      DOUBLE PRECISION ARRAY,AMAX,SAVE
      INTEGER IK,JK,NDEGRE,I,J,K,L,MAXPRS,MAXDGE

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION ARRAY(1:MAXDGE,1:MAXDGE),IK(1:MAXDGE),JK(1:MAXDGE)
      COMMON/BLK1/ARRAY

      WRITE (*,*) 'Determining the inverse of the coefficient matrix.'
      WRITE (*,*)
      DO 100 K=1,NDEGRE

C
C     DETERMINE LARGEST ELEMENT ARRAY(I,J) IN MATRIX
C

       AMAX = 0.D0
```

```
30    DO 10 I=K,NDEGRE
        DO 20 J=K,NDEGRE
          IF (ABS(AMAX)-ABS(ARRAY(I,J)).GT.0.D0) GO TO 20
          AMAX=ARRAY(I,J)
          IK(K)=I
          JK(K)=J
20      CONTINUE
10    CONTINUE

C
C     INTERCHANGING ROWS AND COLUMNS TO PLACE AMAX ON DIAGONAL
C

      IF (AMAX.NE.0.D0) GO TO 40
      WRITE (*,*) 'Matrix is singular, cannot determine the inverse.'
      WRITE (*,*) 'Program terminating.'
      WRITE (*,*)
      STOP
40    I=IK(K)
      IF (I-K.LT.0) GO TO 30
      IF (I-K.EQ.0) GO TO 50
      DO 60 J=1,NDEGRE
        SAVE=ARRAY(K,J)
        ARRAY(K,J)=ARRAY(I,J)
        ARRAY(I,J)=-1.D0*SAVE
60    CONTINUE
50    J=JK(K)
      IF (J-K.LT.0) GO TO 30
      IF (J-K.EQ.0) GO TO 70
      DO 80 I=1,NDEGRE
        SAVE=ARRAY(I,K)
        ARRAY(I,K)=ARRAY(I,J)
        ARRAY(I,J)=-1.D0*SAVE
80    CONTINUE

C
C     DETERMINE INVERSE MATRIX ELEMENTS
C

70    DO 90 I=1,NDEGRE
        IF (I-K.EQ.0) GO TO 90
        ARRAY(I,K)=-1.D0*ARRAY(I,K)/AMAX
90    CONTINUE
      DO 110 I=1,NDEGRE
        DO 120 J=1,NDEGRE
          IF (I-K.EQ.0) GO TO 120
          IF (J-K.EQ.0) GO TO 120
          ARRAY(I,J)=ARRAY(I,J)+ARRAY(I,K)*ARRAY(K,J)
120     CONTINUE
110   CONTINUE
      DO 130 J=1,NDEGRE
        IF (J-K.EQ.0) GO TO 130
        ARRAY(K,J)=ARRAY(K,J)/AMAX
130   CONTINUE
```

```
      ARRAY(K,K)=1.D0/AMAX
100   CONTINUE

C
C     RESTORE ORIGINAL MATRIX ROW/COLUMN ORDERING
C

      DO 140 L=1,NDEGRE
        K=NDEGRE-L+1
        J=IK(K)
        IF (J-K.LE.0) GO TO 150
        DO 160 I=1,NDEGRE
          SAVE=ARRAY(I,K)
          ARRAY(I,K)=-1.D0*ARRAY(I,J)
          ARRAY(I,J)=SAVE
160     CONTINUE
150     I=JK(K)
        IF (I-K.LE.0) GO TO 140
        DO 180 J=1,NDEGRE
          SAVE=ARRAY(K,J)
          ARRAY(K,J)=-1.D0*ARRAY(I,J)
          ARRAY(I,J)=SAVE
180     CONTINUE
140   CONTINUE
      RETURN
      END
```

UNSCAL.FOR

```
C    SUBROUTINE UNSCAL
C
C    PURPOSE
C
C     RESTORES SCALED (NORMALIZED) ESR SPECTRUM TO ORGINAL SIZE
C     SIZE   AFTER   POLYNOMIAL   FIT   AND   RESCALES   POLYNOMIAL
COEFFICIENTS
C
C    USAGE
C
C     CALL UNSCAL(NPAIRS,REDX,REDY,NDEGRE)
C
C    DESCRIPTION OF PARAMETERS
C
C     NDEGRE – DEGREE OF POLYNOMIAL + 1
C     POLYN  – COEFFICIENTS FROM POLYNOMIAL FIT
C     REDX   – SCALING (NORMALIZING) FOR INDEPENDENT VARIABLE
C     REDY   – SCALING (NORMALIZING) FOR DEPENDENT VARIABLE
C     I,J    – ITERATION COUNTER
C     MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C     MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C
C    SUBROUTINES REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C      ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE UNSCAL(REDX,REDY,NDEGRE)

      DOUBLE PRECISION REDX,REDY,POLYN
      INTEGER I,NDEGRE,MAXPRS,MAXDGE,J

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION POLYN(1:MAXDGE)
      COMMON/BLK4/POLYN

      WRITE (*,*) 'Scaling the coefficients for output.'
      WRITE (*,*)

C
C     RESCALES POLYNOMIAL COEFFICIENTS
C

      DO 20 I=1,NDEGRE
        POLYN(I)=POLYN(I)*REDY
        DO 30 J=1,I-1
          POLYN(I)=POLYN(I)/REDX
30      CONTINUE
```

```
20   CONTINUE
     RETURN
     END
```

CHEOUT.FOR

```
C    SUBROUTINE CHEOUT
C
C    PURPOSE
C
C     OUTPUTS RESULTS OF POLYNOMIAL FIT TO DATA
C
C    USAGE
C
C    CALL CHEOUT(DATLIS,NDEGRE)
C
C    DESCRIPTION OF PARAMETERS
C
C     DATLIS – ASSIGNED OUTPUT FILE FOR007
C     NDEGRE – DEGREE OF POLYNOMIAL + 1
C     POLYN  – COEFFICIENTS FROM POLYNOMIAL FIT
C     I      – ITERATION COUNTER
C     MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS + 1
C     MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C
C    SUBROUTINES REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C       ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE CHEOUT(DATLIS,NDEGRE)

      INTEGER NDEGRE,I,MAXPRS,MAXDGE
      DOUBLE PRECISION POLYN
      CHARACTER*20 DATLIS

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION POLYN(1:MAXDGE)
      COMMON/BLK4/POLYN

C
C     OUTPUTS RESULTS TO SCREEN
C

      WRITE (*,'(26(/))')
      WRITE (*,1)
1                                                                    FORMAT
('-------------------------------------------------------------------
     1-------------------------------')
      WRITE (*,*) '                    Baseline Correction Results'
      WRITE (*,1)
      WRITE (*,*)
      WRITE (*,*) 'Coefficients of the Chebyschev Polynomial:'
      WRITE (*,*)
```

```
      WRITE (*,*) '[where P(x)=a(0)+a(1)*x+a(2)*x^2+a(3)*x^3+...]'
      WRITE (*,*)
      WRITE (*,*)
      DO 10 I=1,NDEGRE
        WRITE (*,20) 'a(',I-1,') = ',POLYN(I)
20      FORMAT (1P,A3,I2,A4,D17.9)
10    CONTINUE

C
C     OUTPUTS RESULTS TO FILE FOR007
C

      OPEN (UNIT=7,FILE=DATLIS,STATUS='UNKNOWN')
      WRITE (7,1)
      WRITE (7,*) '                    Baseline Correction Results'
      WRITE (7,1)
      WRITE (7,*)
      WRITE (7,*) 'Coefficients of the Chebyschev Polynomial:'
      WRITE (7,*)
      WRITE (7,*) '[where P(x)=a(0)+a(1)*x+a(2)*x^2+a(3)*x^3+...]'
      WRITE (7,*)
      WRITE (7,*)
      DO 40 I=1,NDEGRE
        WRITE (7,30) 'a(',I-1,') = ',POLYN(I)
30      FORMAT (1P,A3,I2,A4,D17.9)
40    CONTINUE
      CLOSE (UNIT=7,STATUS='KEEP')
      RETURN
      END
```

SPECHE.FOR

```
C    SUBROUTINE SPECHE
C
C    PURPOSE
C
C     OUTPUT ESR SPECTRUM AND POLYNOMIAL FIT TO BASELINE
C
C    USAGE
C
C     CALL SPECHE(NPAIRS,NDEGRE,DATCOM)
C
C    DESCRIPTION OF PARAMETERS
C
C     DATCOM – ASSIGNED OUTPUT FILE FOR008
C     XDATA – ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C     YDATA – ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C     NPAIRS – NUMBER OF DATA POINT PAIRS
C     NDEGRE – DEGREE OF POLYNOMIAL + 1
C     POLYN  – POLYNOMIAL COEFFICIENTS
C     TEMP   – CALCULATED DEPENDENT VARIABLE FROM POLYNOMIAL FIT
C     I,J    – ITERATION COUNTER
C     MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS
C     MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C
C    SUBROUTINES REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C       ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE SPECHE(NPAIRS,NDEGRE,DATCOM)

      DOUBLE PRECISION XDATA,YDATA,TEMP,POLYN
      INTEGER NPAIRS,I,J,NDEGRE,MAXPRS,MAXDGE
      CHARACTER*20 DATCOM

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XDATA(1:MAXPRS),YDATA(1:MAXPRS),POLYN(1:MAXDGE)
      COMMON/BLK3/XDATA,YDATA
      COMMON/BLK4/POLYN

      WRITE (*,*) 'Writing the ESR spectrum and the baseline fit file.'
      WRITE (*,*)
C
C     CALCULATED POLYNOMIAL FIT TO BASELINE
C

      DO 30 I=1, NPAIRS
        TEMP=POLYN(NDEGRE)
        DO 20 J=1,NDEGRE–1
```

```
        TEMP=TEMP*XDATA(I)+POLYN(NDEGRE-J)
20     CONTINUE

C
C    OUTPUT ESR SPECTRUM AND BASELINE CORRECTION
C

     OPEN (UNIT=8,FILE=DATCOM,STATUS='UNKNOWN')
       WRITE (8,15) XDATA(I),',',YDATA(I),',',TEMP
15     FORMAT (F18.6,A,F18.6,A,F18.6)
30    CONTINUE
     CLOSE (UNIT=8,STATUS='KEEP')
     RETURN
     END
```

SPECOT.FOR

```
C    SUBROUTINE SPECOT
C
C    PURPOSE
C
C     OUTPUT BASELINE CORRECTED ESR SPECTRUM
C
C    USAGE
C
C     CALL SPECOT(NPAIRS,NDEGRE,DATOUT)
C
C    DESCRIPTION OF PARAMETERS
C
C     DATOUT – ASSIGNED OUTPUT FILE FOR006
C     XDATA – ARRAY OF DATA POINTS FOR INDEPENDENT VARIABLE
C     YDATA – ARRAY OF DATA POINTS FOR DEPENDENT VARIABLE
C     NPAIRS – NUMBER OF DATA POINT PAIRS
C     NDEGRE – DEGREE OF POLYNOMIAL + 1
C     POLYN  – POLYNOMIAL COEFFICIENTS
C     TEMP   – CALCULATED DEPENDENT VARIABLE FROM POLYNOMIAL FIT
C     I,J    – ITERATION COUNTER
C     MAXPRS – MAXIMUM NUMBER OF DATA POINT PAIRS
C     MAXDGE – MAXIMUM DEGREE OF POLYNOMIAL + 1
C
C    SUBROUTINES REQUIRED
C
C     NONE
C
C    NOTE
C
C     THE PROGRAM REQUIRES A DATA FILE NAMED DIMSIZES.DAT TO
C       ALLOCATE ARRAY DIMENSIONS DURING COMPILATION.
C

      SUBROUTINE SPECOT(NPAIRS,NDEGRE,DATOUT)

      DOUBLE PRECISION XDATA,YDATA,TEMP,POLYN
      INTEGER NPAIRS,I,J,NDEGRE,MAXPRS,MAXDGE
      CHARACTER*20 DATOUT

$INCLUDE:'DIMSIZES.DAT'

      DIMENSION XDATA(1:MAXPRS),YDATA(1:MAXPRS),POLYN(1:MAXDGE)
      COMMON/BLK3/XDATA,YDATA
      COMMON/BLK4/POLYN

      WRITE (*,*) 'Writing the corrected ESR spectrum file.'
      WRITE (*,*)
C
C     DETERMINE BASELINE CORRECTION
C

      DO 10 I=1, NPAIRS
        TEMP=POLYN(NDEGRE)
        DO 20 J=1,NDEGRE–1
```

```
        TEMP=TEMP*XDATA(I)+POLYN(NDEGRE-J)
20      CONTINUE
        YDATA(I)=YDATA(I)-TEMP
10    CONTINUE

C
C     OUTPUT CORRECTED ESR SPECTRUM (XDATA AND YDATA)
C

      OPEN (UNIT=6,FILE=DATOUT,STATUS='UNKNOWN')
      DO 30 I=1,NPAIRS
        WRITE (6,15) XDATA(I),',',YDATA(I)
15      FORMAT (F18.6,A,F18.6)
30    CONTINUE
      CLOSE (UNIT=6,STATUS='KEEP')
      RETURN
      END
```

DIMSIZES.DAT

```
C     THIS FILE CONTAINS THE DIMENSIONING DATA FOR THE ESR BASELINE
C     CORRECTION ROUTINE.
C
C     USERS WHO WISH TO ALTER THE ARRAY DIMENSIONS USED IN THE
C     ROUTINE SHOULD CHANGE THE NUMBERS IN THE FOLLOWING CODE:
C
C     MAXPRS = MAXIMUM NUMBER OF X/Y DATA POINT PAIRS +1
C     MAXDGE = MAXIMUM ORDER OF POLYNOMIAL + 1
C
C     THE PROGRAM MUST BE COMPILED AND LINKED AFTER ALTERATION
C     USING THE COMPILE.BAT PROGRAM.
C

      PARAMETER (MAXPRS=2001)
      PARAMETER (MAXDGE=31)
```

BASELINE.BAT

```
ECHO OFF
ECHO Initializing Files
IF NOT EXIST %1.DAT GOTO :MSG1
IF EXIST %1.OUT  GOTO :MSG2
IF EXIST %1.LIS  GOTO :MSG2
IF EXIST %1.CHK  GOTO :MSG2
COPY %1.DAT FOR005
ECHO Running ESR Baseline Correction on
ECHO Data File %1.DAT
ECHO With ESRBASE.EXE
ECHO Press [Control C] to quit or any other key to continue
PAUSE
ESRBASE
ECHO Press any key to continue
PAUSE
ECHO Renaming Output Files
DEL FOR005
RENAME FOR006 %1.OUT
RENAME FOR007 %1.LIS
RENAME FOR008 %1.CHK
DIR %1.*
GOTO :END
:MSG1
ECHO %1.DAT File was not found – File does not exist or was not specified
GOTO :END
:MSG2
ECHO %1.OUT  %1.LIS  %1.CHK  Already exist – Rename the %1.DAT file
:END
```

INSTALL.BAT

```
ECHO OFF
ECHO Copying the source code and utility files to %1
SET TODIR= %1
COPY INVERS.FOR  %TODIR%
COPY ESRBASE.FOR %TODIR%
COPY CHEFIT.FOR  %TODIR%
COPY CHEOUT.FOR  %TODIR%
COPY SCALE.FOR   %TODIR%
COPY SPECIN.FOR  %TODIR%
COPY SPECOT.FOR  %TODIR%
COPY SPECHE.FOR  %TODIR%
COPY UNSCAL.FOR  %TODIR%
COPY COMPILE.BAT %TODIR%
COPY DIMSIZES.DAT %TODIR%
SET TODIR=
CD %1
COMPILE
```

COMPILE.BAT

```
ECHO OFF
ECHO Compiling source code
fl /c /FPi87 /G2 INVERS.FOR
fl /c /FPi87 /G2 ESRBASE.FOR
fl /c /FPi87 /G2 CHEFIT.FOR
fl /c /FPi87 /G2 CHEOUT.FOR
fl /c /FPi87 /G2 SCALE.FOR
fl /c /FPi87 /G2 SELECT.FOR
fl /c /FPi87 /G2 SPECIN.FOR
fl /c /FPi87 /G2 SPECOT.FOR
fl /c /FPi87 /G2 SPECHE.FOR
fl /c /FPi87 /G2 UNSCAL.FOR
ECHO linking object code to create executable ESRBASE.EXE
LINK ESRBASE INVERS CHEFIT CHEOUT SCALE SELECT SPECIN SPECOT
SPECHE UNSCAL /E;
ECHO deleting object code
DEL *.OBJ
ECHO Installation complete
```